TDB-ACC-NO: NN77054530

DISCLOSURE TITLE: Interrupt Processing in a Structure Programming Environment on a Stack Microprocessor. May 1977.

PUBLICATION-DATA: IBM Technical Disclosure Bulletin, May 1977, US

VOLUME NUMBER: 19

ISSUE NUMBER: 12

PAGE NUMBER: 4530 - 4531

PUBLICATION-DATE: May 1, 1977 (19770501)

CROSS REFERENCE: 0018-8689-19-12-4530

DISCLOSURE TEXT:

2p. The occurrence of interrupts in a structured programming environment represents, on current processors, considerable difficulties and is, at best, an ill-defined state of affairs. That is, the interrupt, because of processor architecture, does not fit well into the concepts of structured programming. Its activation is not well predicted and, when it occurs, causes an ungentle disruption of current operations. Most texts on structured programming ignore the subject.

- A further problem with interrupt handling, and especially interrupt handling in a structured programming environment, is the uncertainty involved in the correctness of the instruction address (as derived by the interrupt type). That is, the current processors are incapable of detecting invalid target address. This is especially a problem in programming systems which must dynamically alter interrupt target addresses to account for relocatability, as well as for other (system-related) purposes.

- This article describes a uniform method for invoking interrupt processing in a structured programming environment using a stack microprocessor. The method comprises the steps of:

(1) recognizing the interrupt type,
(2) selecting the predefined process for servicing the interrupt,
(3) encapsulating into the stack the current process utilizing the identical mechanisms invoked for program "calls",

(4) allocating a new level in the stack for the interrupt process invoked, and

(5) executing the first instruction of the invoked process, if said first instruction is of the predefined type.

- The point of novelty resides in the overlapping of the testing and execution of the first instruction of the "invoked process". This permits a defined entry point to a structured program process to be ascertained and checked by microprocessor hardware at the time of invocation of the process due to the occurrence of the hardware event. 4785311

- In the prior art, the use of program status words in OS/360 for saving the current register-contents of an interrupted process serves much the same function as encapsulation. Also in OS/360 "target addresses" to which control is intended to pass from one process to another have been tested by embedding FETCH and TEST instructions prior in sequence to a BRANCH and LINK. Lastly, U. S. Patent 3,348,211 shows the overlapping of instruction sequence processing based on embedding a TRANSFER instruction to a second sequence a predetermined number of instruction steps prior to the end of a first sequence on a CPU having overlapping memory cycles. Relatedly, U. S. Patent 3,956,735 shows a microprocessor having a mechanism for CALLING selected subsequences, passing control to the called sequence, and returning control when through.

However, there is no teaching of CONCURRENT EXECUTION of a test and transfer instruction.

SECURITY: Use, copying and distribution of this data is subject to the restictions in the Agreement For IBM TDB Database and Related Computer Databases. Unpublished - all rights reserved under the Copyright Laws of the United States. Contains confidential commercial information of IBM exempt from FOIA disclosure per 5 U.S.C. 552(b)(4) and protected under the Trade Secrets Act, 18 U.S.C. 1905.

COPYRIGHT STATEMENT: The text of this article is Copyrighted (c) IBM Corporation 1977. All rights reserved.